Compiling Image Processing Applications for Many-Core Accelerators

Pierre Guillou

Compiling Image Processing Applications for Many-Core Accelerators

Pierre Guillou – CRI MINES ParisTech, PSL Research University

Image Processing

image analysis: detect geometrical structures in an image

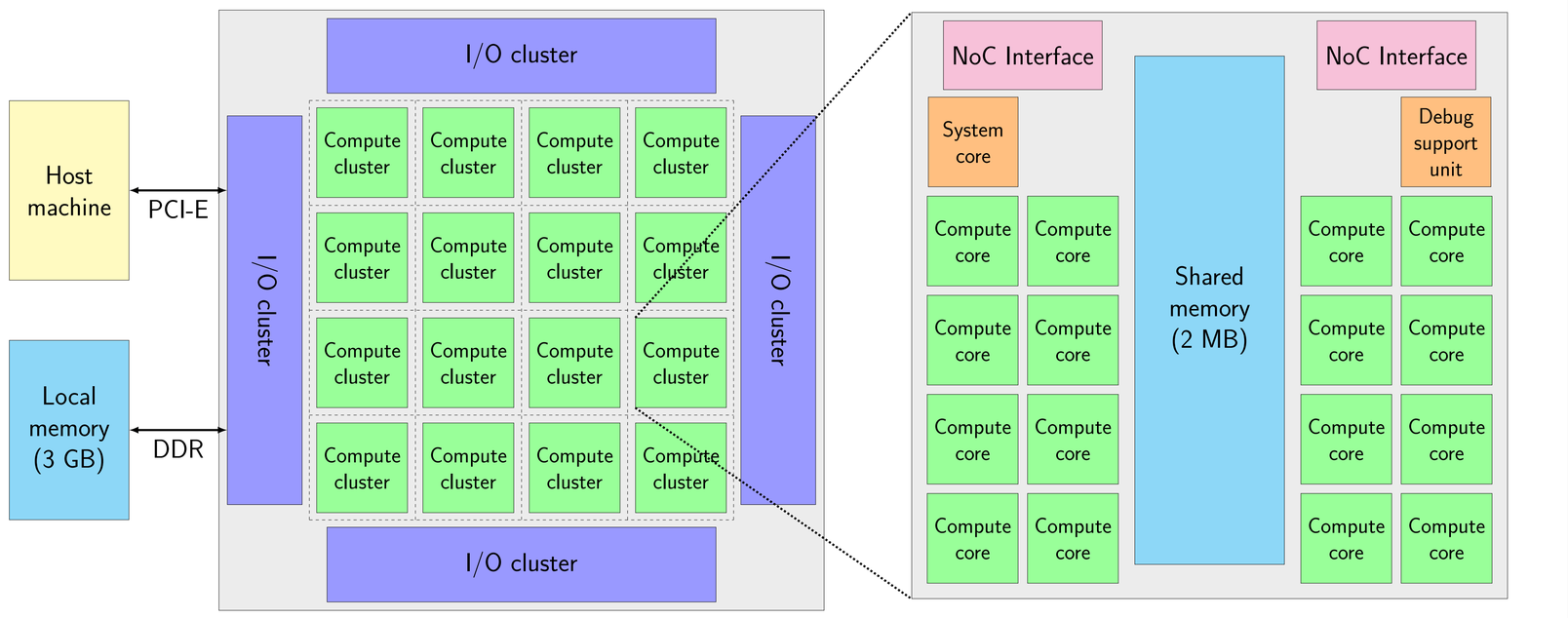
mathematical morphology: image analysis theory and technique based on lattices theory

Mathematical Morphology Base Operators

- arithmetic operators
  - unary (pixel $\otimes$ parameter, 1 input image)
  - binary (pixel $\otimes$ pixel, 2 input images)
  - $+$ $-$ $\times$ $\div$ min max $=$ & | $\sim$
- morphological operators
  - stencils
  - neighbor selection + min/max/avg
- reduction operators
  - global max/min/sum
- other operators
  - threshold, mask, log2, ...

$\Longrightarrow$ *Sigma-C* agent library

The MPPA-256 Chip

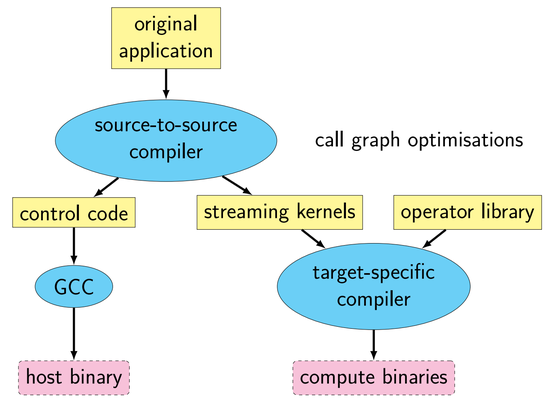


Example: Licence Plate Extraction

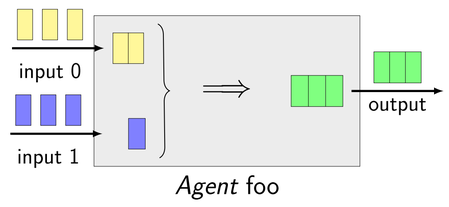
Input

Output

Compilation Chain



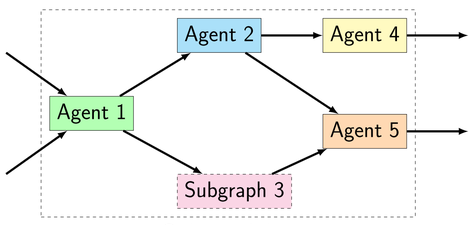
Sigma-C, a *Dataflow* Programming Language



Agent foo

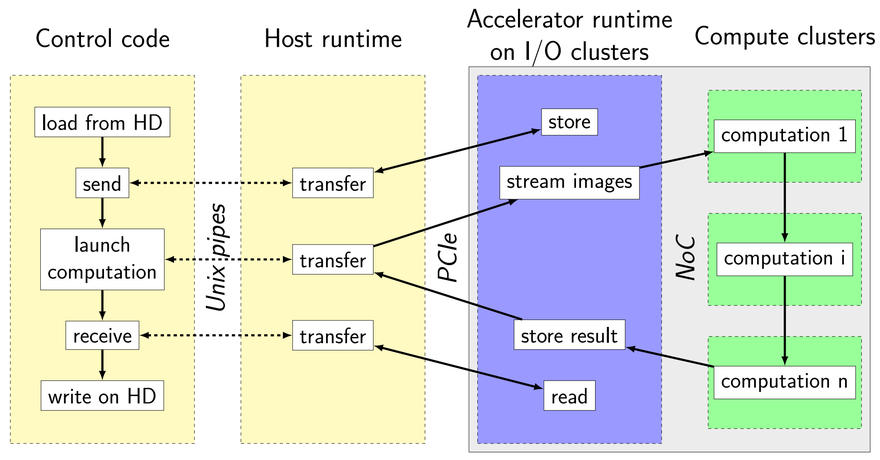
```
agent foo() {
  // describe agent interface
  interface {
    in<int> input0, input1;
    out<int> output;
    // declare the state machine
    spec{input0[2], input1, output[3]};
  }
  // loop over the state
  void start() exchange (input0 inp0[2], input1 inp1,
                         output outp[3]) {
    outp[0] = inp0[0];
    outp[1] = inp1;
    outp[2] = inp0[1];
  }
}
```

```
subgraph bar() {
  // describe subgraph interface
  interface { /* ... */ }
  map {
    // instanciate agents
    agent a1 = new Agent1();
    agent a3 = new Subgraph3(); // ...
    // connect agents to subgraph interfaces
    connect (input0, a1.input0);
    connect (a5.output, output1); // ...
    // connect agents
    connect (a1.output0, a2.input);
    connect (a3.output, a5.input1); // ...
  }
}
```



Subgraph bar

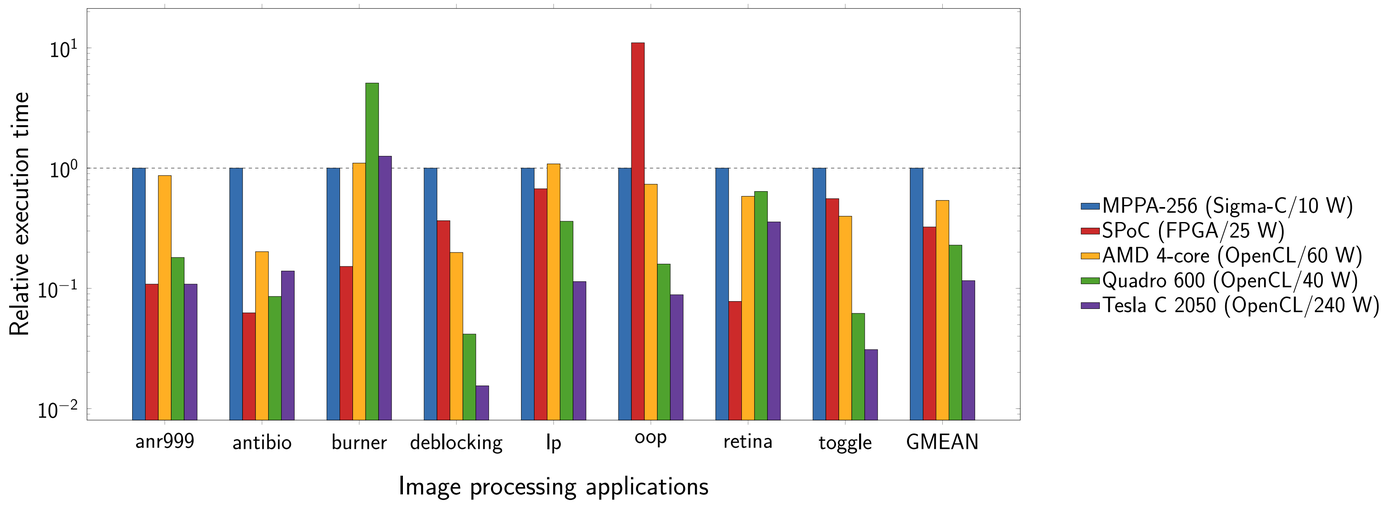
Runtime Environment

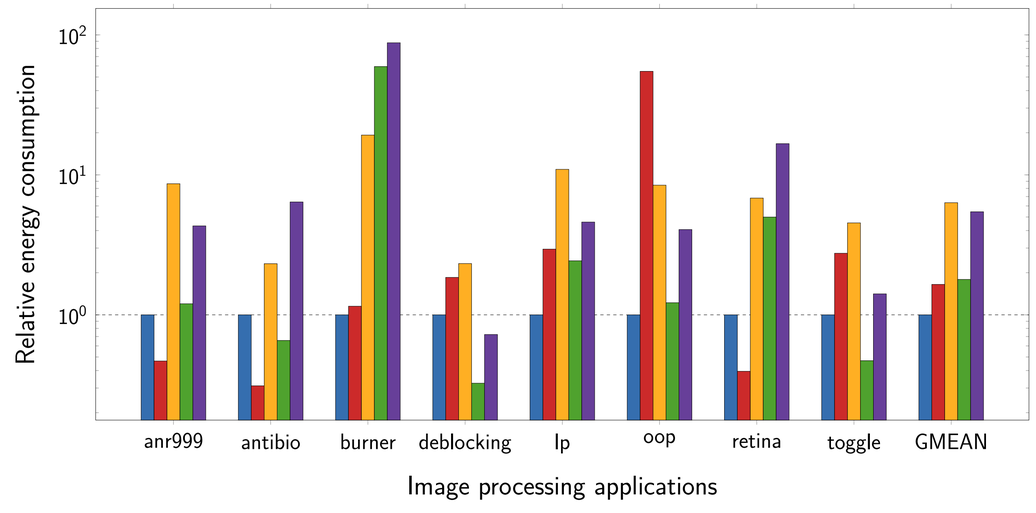


Optimisations

- unrolling of converging loops
- arithmetic operators aggregation
- generation of kernel-specific convolutions
- data parallelization for compute-intensive operators

Results: Execution Times and Energy Consumption (MPPA-256 = 1, lower is better)





Future Work

- Other programming models:
  - Pthreads/OpenMP on compute clusters, communication library between clusters
  - OpenCL via local memory pagination
- Improve data-parallelism to take better advantage of the curent architecture
- Implement more complex algorithms: watershed, arrow, labelling, minima, ...

References

Pierre Guillou, Fabien Coelho, and François Irigoin.
Automatic Streamization of Image Processing Applications.
The 27th International Workshop on Languages and Compilers for Parallel Computing (LCPC), 2014.
Available at `http://www.cri.ensmp.fr/classement/doc/A-570.pdf`.